IN THE CLAIMS

Please cancel claims 2-4, 17, 18, 21, 28, and 30-60 without prejudice.

Please amend claims 1, 14, 20, and 24 as follows:

1. (Currently Amended) A method for processing audio data, comprising:

training time-delay neural network (TDNN) classifiers using **a time-delay neural network that uses** a first layer followed by a second layer having a nonlinearity;

using discriminatively-trained classifiers that are time-delay neural network ~~(TDNN)~~ classifiers to produce a plurality of anchor models;

applying the plurality of anchor models to the audio data;

obtaining a preliminary output of the plurality of anchor models from ~~a~~ **the** time-delay neural network **during training of the TDNN classifiers** before **final nonlinearities are applied by** the second layer ~~having the nonlinearity is applied~~ in order to generate an output of the plurality of anchor models;

normalizing the output of the plurality of anchor models to generate a normalized output of the plurality of anchor models;

mapping the normalized output of the plurality of anchor models into frame tags; and

producing the frame tags.

2. (Canceled)

3. (Canceled)

4. (Canceled)

5. (Previously Presented) The method as set forth in claim 1, further comprising training the TDNN classifier using cross entropy.

6. (Original) The method as set forth in claim 1, further comprising pre-processing the audio data to generate input feature vectors for the discriminatively-trained classifier.

7. (Original) The method as set forth in claim 1, further comprising normalizing a feature vector output of the discriminatively-trained classifier.

8. (Original) The method as set forth in claim 7, wherein the normalized feature vectors are vectors of unit length.

9. (Original) The method as set forth in claim 1, further comprising:

accepting a plurality of input feature vectors corresponding to audio features contained in the audio data; and

applying the discriminatively-trained classifier to the plurality of input feature vectors to produce a plurality of anchor model outputs.

10. (Original) The method as set forth in claim 1, wherein the mapping comprises:

clustering anchor model outputs from the discriminatively-trained classifier into separate clusters using a clustering technique; and

associating a frame tag to each separate cluster.

11. (Original) The method as set forth in claim 10, further comprising applying temporal sequential smoothing to the frame tag using temporal information associated with the anchor model outputs.

12. (Original) The method as set forth in claim 1, further comprising:

training the discriminatively-trained classifier using a speaker training set containing a plurality of known speakers; and

pre-processing the speaker training set and the audio data in the same manner to provide a consistent input to the discriminatively-trained classifier.

13. (Original) A computer-readable medium having computer-executable instructions for performing the method recited in claim 1.

14. (Currently Amended) A computer-implemented process for processing audio data, comprising:

applying a plurality of anchor models to the audio data, the plurality of anchor models comprising discriminatively-trained classifiers of a convolutional neural network that were previously trained using a training technique using a first layer followed by a second layer having a nonlinearlity;

obtaining a preliminary output of the plurality of anchor models from the convolutional neural network **during training of the discriminatively-trained classifiers** before **final nonlinearities are applied by** the second layer ~~having the nonlinearity is applied~~ in order to generate a modified feature vector output;

normalizing the modified feature vector output to generate normalized anchor model output;

mapping the normalized anchor model output into frame tags; and

producing the frame tags.

15. (Original) The computer-implemented process of claim 14, wherein the training technique employs a cross-entropy cost function.

16. (Original) The computer-implemented process of claim 14, wherein the training technique employs a mean-square error metric.

17. (Canceled)

18. (Canceled)

19. (Previously Presented) The computer-implemented process of claim 14, wherein normalizing further comprises creating a modified feature vector output having unit length.

20. (Currently Amended) A method for processing audio data containing a plurality of speakers, comprising:

training time-delay neural network (TDNN) classifiers using **a time-delay neural network that uses** a first layer followed by a second layer having a nonlinearity;

using the TDNN classifiers to produce a plurality of anchor model outputs;

applying the plurality of anchor models to the audio data;

obtaining a preliminary output of the plurality of anchor models from a time-delay neural network **during training of the TDNN classifiers** before **final nonlinearities are applied by** the second layer ~~having the nonlinearity is applied~~ in order to generate an output of the plurality of anchor models;

normalizing the output of the plurality of anchor models to generate a normalized output of the plurality of anchor models;

mapping the normalized output of the plurality of anchor models into frame tags; and

constructing a list of start and stop times for each of the plurality of speakers based on the frame tags;

wherein discriminatively-trained classifiers were previously trained using a training set containing a set of training speakers, and wherein the plurality of speakers is not in the set of training speakers.

21. (Canceled)

22. (Original) The method as set forth in claim 20, further comprising normalizing a feature vector output from the convolutional neural network classifier by mapping each element of the feature vector output to a unit sphere such that the feature vector output has unit length.

23. (Original) One or more computer-readable media having computer-readable instructions thereon which, when executed by one or more processors, cause the one or more processors to implement the method of claim 20.

24. (Currently Amended) A computer-readable medium having computer-executable instructions for processing audio data, comprising:

training discriminatively-trained classifiers that are time-delay neural network (TDNN) classifiers in a discriminative manner on a convolutional neural network using a training technique such that the training occurs during a training phase to generate parameters that can be used at a later time by the TDNN classifiers and includes two layers with a first layer including a one-dimensional convolution followed by a second layer having a nonlinearity;

using the TDNN classifiers to produce a plurality of anchor model outputs;

obtaining during training the plurality of anchor model outputs from the convolutional neural network prior to application of **final nonlinearities by** the second layer ~~**having a nonlinearity**~~ to generate a modified plurality of anchor model outputs;

normalizing the modified plurality of anchor model output to generate normalized anchor model outputs; and

clustering the normalized anchor model outputs into frame tags of speakers that are contained in the audio data.

25. (Original) The computer-readable medium of claim 24, further comprising pre-processing a speaker training set during the training and validation phase to produce a first set of input feature vectors for the discriminatively-trained classifier.

26. (Original) The computer-readable medium of claim 25, further comprising pre-processing the audio data during the use phase to produce a second set of input feature vectors for the discriminatively-trained classifier, the pre-processing of the audio data being preformed in the same manner as the pre-processing of the speaker training set.

27. (Original) The computer-readable medium of claim 24, further comprising normalizing the feature vector outputs to produce feature vectors having a unit length.

28. (Canceled)

29. (Original) The computer-readable medium of claim 25, further comprising applying temporal sequential smoothing to the clustering the clustered feature vector outputs to produce the frame tags.

Claims 30-60: Canceled